

10. Leah

Program Name: Leah.java

Input File: leah.dat

Leah is rather fond of expressing numbers in different bases. In particular, Leah has a real affinity for binary numbers. As such, she already has a good understanding of what binary numbers are, and how to read them. In most cases, when generating binary numbers, binary numbers are ordered from least to greatest. For example, the following is a list of the binary representation of the numbers 0 through 7 which are expressed to 3 bits:

0b000,	0b001,	0b010,	0b011,	0b100,	0b101,	0b110,	0b111
↓	↓	↓	↓	↓	↓	↓	↓
0	1	2	3	4	5	6	7

Leah in her Digital Logic class was recently introduced to Gray Codes, which are an alternative way to order binary numbers. Rather than simply adding one to the previous binary number to generate the next binary number, Gray Codes order binary numbers according to the simple principle that no two adjacent numbers can differ by more than a single bit. The following is the order of the first 8 Gray Codes expressed to 3 bits:

0b000,	0b001,	0b011,	0b010,	0b110,	0b111,	0b101,	0b100
↓	↓	↓	↓	↓	↓	↓	↓
0	1	3	2	6	7	5	4

However, generating Gray Codes can be decently difficult to do so by hand. Help Leah by writing her a program that generates Gray Codes for different bit widths.

Input: The first line of input will consist of a single integer n ($1 \leq n \leq 32$) denoting the number of testcases to follow. The next n lines will each contain a single integer w_i ($1 \leq w_i \leq 8$) denoting the width of any given binary number that Leah wants to generate.

Output: For each of Leah's n requests, on their own line, print a space-separated list of the decimal representation of the numbers 0 through $2^{w_i} - 1$ in their Gray Codes ordering.

Sample input:

```
4
3
1
4
2
```

Sample output:

```
0 1 3 2 6 7 5 4
0 1
0 1 3 2 6 7 5 4 12 13 15 14 10 11 9 8
0 1 3 2
```